IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

UTILITY APPLICATION FOR UNITED STATES LETTERS PATENT

Express Mail Label No.	:	T. B. D.
Date Deposited	:	T. B. D.
Attorney Docket No.	:	71044/03786
No. of Dwg. Figs./Sheets	:	29
No. of Claims -		
Independent	:	4
Total	:	31

METHOD AND APPARATUS FOR PROGRAMMING VEHICLE NETWORKS

by

PETER LUHRS
DAVID PASTORE
DARYL POPIG
SEAN TILLINGHAST

Assigned to

WELDON TECHNOLOGIES, INC.

Arter & Hadden, LLP
Attorneys at Law
10 WEST BROAD STREET
COLUMBUS, OH 43215-3422
TELEPHONE (614) 221-3155
FACSIMILE (614) 221-0479

METHOD AND APPARATUS FOR PROGRAMMING VEHICLE NETWORKS

Background of the Invention

Technical Field

The present invention is generally related to methods and systems for programming a controller of a multiplexed vehicle network. More specifically, the present invention is related to a method and apparatus for enabling a human vehicle network designer to design, program, simulate, test and debug the operation of a multiplexed vehicle network controller and associated components.

Background Art

Technology advancement has not significantly improved basic vehicle power systems since the mid-1950's when the automotive industry converted from six volt systems to 12 volt systems. Since then, however, there have been numerous new features and components incorporated into vehicles spurred by availability, user demand, competitive pressures and regulatory requirements. Such new features include comfort and convenience devices, safety enhancements, entertainment sources, navigation and communication devices, engine and transmission management, on-board diagnostic circuits and lighting and annunciation systems unique to mission performance, e.g., emergency vehicle lights and audible alarms.

The number of additional features requiring electrical power is expected to increase leaving manufacturers confronted with the problem of how to accommodate the growing complexity of each vehicle's electrical system. Recent industry estimates predict that the value of electronics in a typical car, which totaled $940 in 1990, will rise to $1,720 per car by 2005.

Today, even without all of the available options, the average passenger vehicle wiring harness totals over a mile in length, weighs in excess of 60 pounds and terminates in over 500 connections. Each vehicle has numerous relays, switches, wiring splices, fuses and other ancillary components. Manufacturing issues include vehicular weight, spatial limitations, style, packaging considerations and the ability to readily incorporate or delete optional features. These limitations and issues are magnified in specialized vehicles, such as, for example, police cars, fire trucks, ambulances, school buses, transit buses, snow removal equipment, street cleaners and, tow trucks.

An additional concern for such specialized vehicles is the need for proper system integration and information sharing between and among the vehicle's sub-systems. Because additional features produce significantly increased demands on a vehicle's limited-capacity electrical system, efficient power management becomes a priority concern. In addition to efficient power management, a vehicle's design must also take into account system quality, reliability, maintainability and cost.

In response to the increasing number of desirable features powered by a vehicle's electrical system, some vehicle manufacturers and after-market component providers have begun to implement vehicle networks to overcome many of the deficiencies of traditional vehicle electrical systems outlined above. Several in-vehicle network protocols are being used, including J1850, Control Area Network ("CAN"), Local Interconnect Network ("LIN"), and Vehicle Multiplexed Network ("V-MUX").

The CAN protocol has the advantage of enabling relatively fast communication. CAN is already a well accepted standard among European automobile manufacturers, and is becoming a commonly used vehicle network in America and the Far East.

The LIN protocol was jointly developed by Audi, BMW, Daimler-Chrysler, Motorola, Volcano Communications Technologies ("VCT"), Volkswagen and Volvo. The LIN specification covers not only the protocol definition, but also the standardization of tool and application programming interfaces. LIN is a relatively low speed network and transmits between 1 K and 20 Kbit per second. First applications based on LIN will appear in cars manufactured in 2001 with three to ten LIN nodes per vehicle. Over the next ten years, vehicles are expected to support up to 20 nodes each.

V-MUX is a proprietary network protocol that operates from 1k to 100k. V-MUX can operate effectively using a single twisted pair of wires having 1 twist per inch.

While each of the current network protocols solves several deficiencies of the prior art vehicle electrical systems, use of such networks results in certain deficiencies:

- In order to operate properly, each vehicle network requires controller programming which must support the configuration of components installed in the vehicle. Presently, the programming of vehicle networks is accomplished utilizing network programming modules that require a user to have specialized knowledge and/or be familiar with ladder logic;
- Debugging a new vehicle controller program using a conventional network programming module is inefficient and time consuming;
- Maintaining vehicle network controller programs written by others is difficult; and
- Specialized knowledge of each protocol is required to competently program a vehicle network controller.

A need therefore exists for a product which addresses the shortcomings of employing the

presently available protocols to control and monitor vehicle options via a vehicle network. A need further exists for a method and apparatus which enables a user to easily program a vehicle network controller using basic logic without requiring specialized knowledge such as a programming language or knowledge of a particular protocol. Yet another need exists for a method and apparatus for programming a vehicle network which enables the network to provide increased functionality when compared with conventional methods and systems.

Summary of the Invention

One aspect of the invention relates to a method for programming at least a portion of a multiplexed vehicle network. The method includes the step of receiving user input via an intuitive graphical user interface. The user input is used to perform the steps of identifying the layout of a vehicle network and defining logical relationships between components of the vehicle network. The method further includes the step of compiling network data based on the layout and logical relationships. The compiled data is stored for later use or for transferring to a vehicle network node.

A second aspect of the present invention relates to an apparatus for programming at least a portion of a multiplexed vehicle network. The apparatus includes a means for receiving user input via an intuitive graphical user interface ("GUI"). The apparatus also includes a means for identifying the layout of a vehicle network based on user input received via the GUI. The apparatus further includes a means for defining logical relationships between components of the vehicle network based on the user input received via the GUI. The apparatus still further includes a means for compiling network data based on the layout and logical relationships, and a means for storing the compiled data

A third aspect of the present invention relates to an apparatus for programming at least a portion of a multiplexed vehicle network comprising a processor and a memory connected to the processor. The memory stores a program to control the operation of the processor, and enables the apparatus to perform the steps of the above described method of the present invention.

A fourth aspect of the invention relates to a computer-readable storage medium encoded with processing instructions. The processing instructions direct a processor to perform the steps of the above described method of the present invention.

A main advantage of the present invention is that a vehicle network can be designed and programmed without requiring a user to possess specialized knowledge.

Another advantage of the present invention is that a vehicle network can be implemented without requiring excessive wiring .

The objects, features and advantages of the present invention are readily apparent from the following description of the preferred embodiments when taken in connection with the accompanying drawings.

Brief Description of the Drawings

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

Figure 1 is a schematic block diagram illustrating the hardware environment of an embodiment of the present invention;

Figure 2 is a block diagram illustrating the primary components of one preferred node of an embodiment of the present invention;

Figure 3 is a block diagram illustrating the preferred protocol of an embodiment of the present invention;

Figure 4 is an elevational view of a display node of an embodiment of the present invention;

Figure 5 is a block diagram illustrating the software architecture of an embodiment of the present invention;

Figure 6A is a computer screen display illustrating a System Designed Settings box having a User form;

Figure 6B is a computer screen display illustrating a System Designed Settings box having an OEM Commands form;

Figure 6C is a computer screen display illustrating a System Designed Settings box having an Environment form;

Figure 6D is a computer screen display illustrating a System Designed Settings box having a Vista form;

Figure 7 is a computer screen display illustrating the V-MUX System Designer main application window;

Figure 8 is a computer screen display illustrating the Properties window for an I/O node;

Figure 9 is a computer screen display illustrating the Properties window for a rear door switch;

Figure 10 is a computer screen display illustrating the Properties window for a water temperature sensor;

Figure 11 is a computer screen display illustrating the Calibration window for an analog input;

Figure 12 is a computer screen display illustrating the Properties window for an output;

Figure 13 is a computer screen display illustrating the Relationships window for the output;

Figure 14 is a computer screen display illustrating the Design window for a system with two Vista display nodes;

Figure 15 is a computer screen display illustrating the Properties window for screen 1 of Vista display node 6;

Figure 16 is a computer screen display illustrating the Properties window for node 5;

Figure 17 is a computer screen display illustrating the Design Tree including a front strobes command;

Figure 18 is a computer screen display illustrating the Properties window for front strobes command;

Figure 19 is a computer screen display illustrating the Design Tree showing emergency master button of the EMM screen;

Figure 20 is a computer screen display illustrating the Properties window for the emergency master button;

Figure 21 is a computer screen display illustrating the properties window for the emergency master screen-link button;

Figure 22 is a computer screen display illustrating the Design Tree including a PWM dome lights button;

Figure 23 is a computer screen display illustrating the Properties window for the dome lights button;

Figure 24 is a computer screen display illustrating the Design Tree including 10 levels associated with the PWM dome lights button;

Figure 25 is a computer screen display illustrating the Properties window for level 1 of the PWM dome lights button; and

Figure 26 is a computer screen display illustrating the V-MUX Downloader Application window.

Detailed Description

Hardware Environment

The present invention preferably operates in the hardware environment illustrated in Figure 1. Preferred personal computer ("PC") 110 is an IBM compatible computer running a Microsoft Windows™ operating system and having a 200 Mhz Pentium-based processor, at least 16MB of RAM and 30MB of disk space. PC 110 includes a stored application entitled "V-MUX® System Designer" which has been developed by Weldon Technologies, Inc. The V-MUX System Designer application enables a human designer to layout and design the control logic for a proprietary V-MUX® vehicle network, as well as test a design and simulate the operation of a V-MUX vehicle network. PC 110 also includes a stored application program entitled "V-MUX® Downloader", also developed by Weldon Technologies, Inc., which enables an operator to download control logic to a V-MUX vehicle network.

It should be noted that although the preferred embodiment of the invention is described as operating with respect to V-MUX vehicle network, the invention is equally well suited to any conventional vehicle network employing geographically distributed nodes.

The V-MUX Downloader application facilitates the programming of at least a portion of vehicle network 112 within a vehicle, such as vehicle 114. Network data defining the layout of vehicle network 120 and logical relationships between components of the vehicle network is transmitted from PC 110 to a node 200 of vehicle network 120 via transceiver 112. Transceiver 112 is a communications device which converts RS-232 data signal levels to RS-485 data signal levels. A standard 9-pin serial cable 116 connects PC 110 to transceiver 112, and a standard 4 pin RS-485 cable 118 connects transceiver 112 to node 200 of V-MUX network 120. While the invention is not limited to any particular communications protocol, RS-232 is the preferred interface for transferring data from a PC due to its acceptance as a standard in the industry. Of course, the Universal Serial Bus ("USB") interface, now becoming widely accepted in the industry, could be used also. RS-485 was selected as the interface for transferring data to/from node 122, in part, because the RS-485 standard supports longer haul communication lines than RS-232. Further, RS-485 allows nodes 200 and 400 to support a single hardware protocol regardless of whether it is downloading data from PC 110 or communicating with other nodes of network 120.

As shown, preferred V-MUX Vehicle network 120 is a peer to peer nodal hierarchy and includes three network nodes: two (2) I/O or "Hercules" nodes 200 and one (1) display or "Vista" node 400. The nodes are connected with a twisted pair of communication wire, and communicate according to RS-485, thereby supporting low-speed communication of typically 9.6 to 100 Kbauds. The V-MUX system performs several tasks including:

- 1. Supplying power to devices such as lights or fans.
- 2. Reading the status of switches and react to them by turning on devices.

- 3. Reading in analog data from several types of sensors. This data can be displayed and/or used to turn on a warning device.

The I/O nodes 200 enable communication with various connected components, and is described more fully with reference to Figure 2. The Vista display node 400 is a Input/Output device that enables an operator to interface with and control network 120. Of course vehicle network 120 is merely one configuration of many possible configurations of vehicle networks, and the invention will operate with a network having more or fewer of either node 200 or node 400. In fact, the present invention can operate in conjunction with a vehicle network which does not include any display nodes 400.

I/O Node

Referring now to Figure 2, the Input/Output capabilities of V-MUX I/O node 020 are illustrated. Node 200 is controlled by Central Processing Unit ("CPU") 210 according to a control program stored in memory 212. The preferred CPU of I/O node 200 is an 8-bit Intel 8051 core, and the preferred memory is flash memory. Of course other CPU choices and/or memory choices may operate equally well. The control program stored in memory 212 utilizes the network data compiled according to the present invention, and the control program may be pre-loaded or loaded along with the network data transmitted to the node in accordance with the present invention.

The I/O node 200 supports sixteen (16) digital signal inputs 214 and four (4) analog inputs 216 for receiving input from components connected to the node. The digital signal inputs 214 are typically connected, for example, to switches, membranes or other input devices that

provide a digital input. The analog inputs 216 are typically used to process real world electrical data , for example, internal temperature, external temperature, oil PSI, oil temperature and engine RPM from analog devices that are connected to the node 200.

The I/O node further supports twenty-four (24) digital power outputs 220 for supplying power to devices connected to the node 200. Digital outputs 220 are used to carry system voltage of typically 12, but in some cases 24, 36, or 42 volts, to power, for example, lamps, gages, LEDs, displays, flood lights, scene lights, throttle and high idle control.

In addition, the I/O node 200 includes two (2) serial communications ports: primary serial communications port 222 and secondary serial communications port 224. Each of the communications ports enable serial data to be sent and received. Depending on the control program of the node, the node may employ any of a number of communications protocols, such as, for example, V-MUX, CAN, J1708, J1587, J1922 or J1939. The secondary serial communications port 224 is often reserved for diagnosing and/or monitoring the engine and/or power train.

V-MUX Message Protocol

Referring now to Figure 3, there is illustrated the V-MUX six-byte protocol of the preferred embodiment of the present invention. Table A sets forth the definition of each data element of the V-MUX Protocol.

Reference Numeral	Bit Width	Data Element
310	8	Tracer Byte
312	4	Start Header

314	10	Command
316	1	On/Off Bit
318	1	Extended Bit
320	16	Data
322	8	Checksum

Table A: V-MUX Protocol

As shown, the V-MUX Protocol begins with an 8-bit tracer byte of zero. When a node is prepared to transmit a message on the vehicle network bus, the node tests the serial (RX) interrupt to determine that the bus is not in use. If the bus is not in use, the node sends out the 8-bit tracer byte as a signal to all nodes that a message is about to be transmitted. If the node and bus support a local echo, the transmitting node may receive the transmitted tracer byte and compare it to determine that no collisions have occurred. If a collision is detected, the transmitting node will abort the transmission and retry the transmission after waiting for a period of time.

After sending the tracer byte, the transmitting node waits a predetermined number of milliseconds to provide a phase delay 330. Following the phase delay, the transmitting node tests the RX interrupt again to determine whether there have been any collisions. As previously described, if a collision is detected, the transmitting node will abort the transmission and retry the transmission after waiting for a period of time.

Following the tracer byte, the transmitting node transmits the remaining five bytes of the

message protocol. Namely, the transmitting node transmits the 4-bit start header, the 10-bit command, the on/off bit, the extended bit, and 16-bits of data and an 8-bit checksum. The start header is always "1010" and represents the beginning of the message. The 10 bit command represents an action requested by the node, or a component connected to the node. For example, the network nodes of an emergency vehicle may be configured such that the state of a certain switch activate a set of emergency flashers. When the activation state of the switch occurs, the node will send out a 10- bit command representing "Emergency Flashers On".

The status bit may be used with certain commands to simulate the functionality of a switch. When the status bit is "0", the command is processed as an off. When the status bit is set to "1", the command is processed as an on.. With respect to the "Emergency Flashers On" command, for example, a status bit set to "1" would cause the emergency flashers to turn on.

In the preferred embodiment, the extended bit is used to define a three-way switch. A node receiving a command with the extended bit set, will reverse the status of the output associated with the command. Of course, in other embodiments the extended bit could be used to enhance the command segment, thereby enabling an additional 1024 commands. The extended bit could further be used in conjunction with the data segment to enlarge the range of data values which may be associated with a command.

If the command has associated data, the data will be transmitted in bytes 3 and 4. One example of a command which utilizes the data bytes is the "Direct PWM Control" command. This command allows a node to be set to a particular PWM percentage. When a Direct PWM Control command is sent, the two data bytes contain the node number channel and the PWM percentage to which the channel will be set. Data element 322 represents an 8-bit checksum. The checksum may be calculated as the sum of bytes 0-4 modulus 256. After transmitting the

entire message, the transmitting node compares the echoed checksum to the calculated checksum to determine whether the transmission was successful. If the compared values do not match, a collision is assumed, and the transmitting node retries the transmission at a later time.

Display Node

Referring now to Figure 4, there is illustrated a front elevational view of a preferred display node 400. Like the I/O node 200, the display node 400 includes a CPU, flash memory, RAM and serial communications port (not illustrated). Display node 400, however, differs from I/O node 200 in that it provides an operator interface for monitoring and controlling vehicle network 120. The illustrated display node 400 includes a display 410 for displaying 40 columns and 16 lines of textual data and/or graphics. Preferably, line 14 of display 410 is reserved for error and other exceptional messages that may be scrolled laterally. Of course, the present invention is not limited to any particular type, size or configuration of the display, and one of ordinary skill will appreciate that any of several conventional displays may be well suited for use with the present invention.

Display node 400 further includes programmable buttons 412-434. Button control is determined based on defined button properties using the V-MUX System Designer software. Buttons 412-434 may be programmed to turn a component on or off, step through a series of preprogrammed settings (e.g. stepping through light settings in 10% increments, effectively creating a dimmer switch), or change the presentation of display 410. Further, button delays may be programmed for buttons 412-434 to enable a command to be processed a specified time after a button is activated.

Buttons 412-426 are numbered buttons which can be programmed for specific tasks depending on options presented on display 410. Buttons 428-434 are programmed to operate regardless of the content of display 410.

Specifically, "HOME" button 428 is preferably programmed to cause display 410 to display a main menu of options. "PRK OVR" or Park Override button 430 is preferably programmed to cause the vehicle network to override the "Park" command which is typically transmitted when the vehicle is placed in "Park". Properly programmed button 430 is useful, for example, when a vehicle operator wishes to preserve the configuration of lights even after the vehicle has come to a stop at the scene of an accident.

"EMR MST" or Emergency Master button 432 is preferably programmed to act as the master emergency light switch. In other words, button 432 enables an operator to turn on and off an entire set of emergency lights. "MOD" or Mod Power button 434 is used to meet ambulance requirements to have a separate button for module power, this button can be programmed to turn off all or some off the devices powered by the network.

Further, display node 400 includes a piezoelectric speaker 436 mounted behind a corresponding grid in the casing of the node 400. Of course, the grid is preferably weather resistant due to the harsh environments in which emergency vehicles often operate.

Display node 400 may be programmed using the V-MUX System Designer software to present several different types of screens, including, for example, menus, information screens and diagnostic screens. A menu screen enables the user to use all or some of buttons 412-426 to control aspects of the vehicle network identified next to each respective button. An information screen presents a set of information to the user, typically in response to input from the user. Finally, diagnostic screens display diagnostic information to assist a user to determine a failure

of one or more components in the vehicle network. Of course, menus and other screens may be nested several layers deep.

Software Environment

Referring to Figure 5, there is illustrated the relationship between software components used by the present invention. Generally, the V-MUX System Designer application 510 enables a user to define components and command which may be used with a vehicle network, layout or configure a vehicle network and define relationships between components of a vehicle network. The V-MUX System Designer receives design data from a user via an intuitive graphical user interface ("GUI"), compiles the design data and outputs compiled data, which when interpreted by a control program of a network node, determines the functionality of the components of a vehicle network.

The compiled data is used as input to the V-MUX Downloader application 512. The V-MUX Downloader application 512 operates in conjunction with transceiver 112, as described with reference to Figure 1, to program a network node to operate according to the vehicle network design provided to the V-MUX System Designer. The V-MUX Downloader 512 causes the compiled data to be transmitted to a node of the vehicle network where it is stored in flash memory and referenced by a network node control program 514.

Network node control program 514 is code that is embedded in the persistent memory of a network node, and it controls the operation of the node, in part, based on the compiled data. While the best mode of the present invention is directed to the generation of the compiled data, it should be recognized that the V-MUX System Designer could also generate the control program, and that the V-MUX Downloader could download the control program to a network node in

addition to the compiled data.

V-MUX System Designer

Upon executing the V-MUX System Designer application for the first time, the application displays a System Designer Settings box as shown in Figure 6A. In the "Settings" tree, the user may select "Environment", "OEM Commands," "User" or "Vista" to alter settings associated with each respective selection. The default selection is "User", and Figure 6A shows the "User" settings form. The user may complete this form with appropriate company and user information. This information is used on reports and other areas throughout the V-MUX System Designer.

Figure 6B shows the "OEM Commands" settings form of the System Designer Settings box. There are ten customizable OEM commands. That can be used to control unique devices for which the V-MUX System Designer does not have a pre-existing command. The user can provide a unique name for each customized command. Because these settings apply to the System Design Global, the OEM commands should be used sparingly. In most cases there will be a command suitable for the user's needs.

Figure 6C shows the "Environment" settings form of the System Designer Settings box. The environment settings allow the user to automatically increment the version of the design the user is working on when saving. This means that every time a design is opened and a change is made, a version number associated with the design in the System Properties window will increment by one number. The V-MUX System Designer also allows the user to select whether a prompt is displayed upon saving. The "remember child window positions" option enables the

user to position the System Designer windows in preferred locations for use during subsequent sessions.

Figure 7 illustrates the main V-MUX System Designer Application window. By selecting the "Blank Page" icon on the upper left of the screen, the user can generate a new system and create three child windows: Design, Relationships, and Properties windows. These three windows may be used together to create a vehicle application. The Design window shows a Windows Explorer™ style hierarchy of a V-MUX vehicle network. In the Design window, each node can be exploded to reveal its inputs and outputs.

The Properties window displays information relating to a highlighted selection. The user can modify the information by editing the fields of the properties window. For example, by selecting on the "Sample System" at the top of the Design tree, the Properties window on the right side of the screen displays information associated with what is selected in the Design tree. The user can fill out the properties thereby defining portions of the System. Table B, below describes each property that may be defined.

Property	Property Description
Name	This property defines the default name of the design. The design file name is based on this field.
Vehicle Type	This property may have one of a set of values selected from a drop down list of options
Vehicle No	In some cases, this property may be the same as Name. Where applicable, a vehicle number may be stored with the design
Version	This property identifies the revision level or version of the design. This number may be automatically incremented if the user chooses that option in the user settings
System Voltage	This property defines the appropriate voltage for the vehicle's system
Alternator Output	This property identifies the total alternator(s) capacity in amps.
Non-V-MUX Loads	This property identifies the load value in amps that will not be multiplexed.

Load Shed Override	This property defines a command that is transmitted over the network to override a load shed command.
Analog Trigger Delay	This value, from 0 – 120, will delay the analog triggers at startup, up to 120 seconds. Delaying the analog triggers will prevent things like low voltage alarms from going off while cranking the vehicle. A value of 120 is recommended as the default. This value may be decreased

Table B: V-MUX Design Properties

Along the left side of the System Designer Main window, there is a tool bar that consists of several icons. The top icon 710 is a blue Weldon logo. Selecting icon 710 docks the tool bar at the top of the screen. Icon 710 acts as a toggle selecting it a second time causes the tool bar to move back. Icon 712 is a minus symbol. It will collapse the node tree from an exploded view. Icon 714 represents an I/O node. Selecting icon 714 adds a Hercules I/O node to the Design Tree. Icon 716 is a Vista display Node. Selecting icon 716 causes a Vista display to be added to the Design Tree. Icon 718 is a Vacuum Fluorescent Display ("VFD"). Selecting icon 718 adds a VFD to the Design Tree. Every Design Tree must have an I/O node selected in order to add a VFD to the Design. Icon 720 allows a screen to be added to a Vista display. Icon 720 may only be selected while working on a Vista to add Screens. Icon 722 is the command set icon. This icon may be used to add command sets to a Vista display. Icon 724 is used to add a library to the design. Libraries are used for custom features and will lock out some outputs in the design from being used. Icon 726 allows a node to be removed from the Design Tree. Icon 728 is the enslave tool. This tool is used to enslave two or more outputs together. This feature is useful for any device that draws more then 10.5 amps. Icon 730 releases enslaved outputs. To free an output

the output that has been enslaved must be selected, followed by a selection of Icon 730. Icon 732 is the compile tool selecting icon 732 compiles the data described.

Referring now to Figure 8, there is illustrated the properties window 800 for an I/O node. The properties window 800 of Figure 8 is displayed upon selecting an I/O node, such as Node 1, in the Design Tree and has two columns: column 810 is the title of the item, column 812 is the description of the item. The "Name" property displays the name of the node; the name cannot be changed. The "Node Type" property displays the type of node the user selected when this node was added to the Design tree. This property cannot be changed. To change a Node Type the node must be deleted, and a new style node must be added. The "Load" property displays the electrical load in amps that the node has been assigned. The Load is a running total of the output properties of the node. Each I/O node is capable of 110 amps. "X Location" property indicates the location of the node in the vehicle. The selections are Left, Center and Right. The "Y Location" further indicates the location of the node in the vehicle and may be set to front, mid-front, middle, mid-rear or Rear. The X and Y Locations will be used together to determine the location of the node in your vehicle. This position is utilized by the Node Layout Report, which produces up to a 'C' sized drawing of your design layout.

The nodes in the Design Tree have inputs and outputs that can be revealed by clicking on the plus sign to the left of the node. For example, clicking on the plus sign to the left of the input icon reveals the inputs. One example input may be a door rear switch. If the user selected that input, the Properties window would change to reveal the properties for the input, as shown in Figure 9. The "Name" property ("Door Rear") can be set to whatever the user feels is appropriate for naming the input. The "Type" property allows the user to indicate whether a

switch is momentary or latching. See the switch table C below for a detailed explanation of momentary vs. latching.

The "Command" property is one of the most important properties. This property actually allows the user to assign a command to an input from the command database. By clicking on the right hand column, a button will appear enabling access to the Command Database. Any command chosen from the command database becomes the default Name of the input. As previously discussed, the name can be changed to something else, but it is recommend to keep the default name. The command assigned to the name property may be used in Diagnostic Utility for troubleshooting. The "On State" property displays the condition that will change the switch status. In this example, 12v or ground. The "Three Way" property is a very powerful feature. The user may set this property to true to control a device from more than one location with the same command. The "Pin" property is the number of that this channel resides at in the V-MUX connector.

Input Overview

In both the Design Tree and the Relationship window, an analog input will have a sine wave icon and a digital input will have a square wave icon. A digital input is used to detect the status of a switch and is either a zero or a one. There are three types of digital inputs: bi-directional, +V and ground. A bi-directional input can be switched by either +V or ground. +V can only switch by +Battery voltage input. A ground input can only be switched by a ground source. Analog inputs are used to read variable voltage from devices such as a pressure transducer or temperature sensor. In the preferred embodiment, it is important to ensure that an attached analog device has a scale inside the 0-5 volt range. The analog inputs of an I/O node

are limited to 0-5 volts. Further, defining all the system inputs for each node in your vehicle is a good approach before trying to assign relationships to the outputs.

Vehicle Device	Type Property	Reason
OEM Horn Ring	Latching	The horn should only stay on when holding the switch down. Defining this as a momentary input, the horn would stay on when you pressed it once.
Rocker Switch – Momentary Type	Momentary	Assigning this switch as momentary tells V-MUX to latch the output ON even though the switch does not latch closed.
Compartment door switch	Latching	A door switch is a latching switch, whether it is magnetic or spring loaded.
Problem	**Cause**	**Reason**
Output connected to a latched switch comes on if I turn the switch on then off	Type Property is Momentary	The Momentary property is assigned to a latched switch
Output connected to a physical momentary switch – Goes on then off when I release the switch.	Type Property is Latched	A Latched property has been assigned to a Momentary switch.

Table C: Switch Table

Referring now to Figure 12, there is illustrated a Properties window for an output selected in the Design window. The properties of an output are similar to the properties for other devices. The "Name" property field enables a descriptive tag to be associated with the output device.

The "Device" property identifies the type of output device and is defined and based on types listed in a drop down box list. The "Volts" property reflects the voltage assigned to the Device and is fixed according to the System Properties. The "Wattage" property reflects the device load in watts. The "Amps" property represents the ampether of the device. Entering a

number in Wattage or Amps, will automatically calculate the other. The "Permanently On" property enables the user to have an output turned on when ever the system is powered, by setting this property to true. The "Flashing" property may be set to true to flash or rapid fire the output. The "Flash Phase" property may be set to A or B depending on the desired flash phase. The "Flash Rate" property sets the speed of any I/O node to flash. The "PWM" property sets the pulse width modulation duty cycle for the output. This is only available on physical outputs 1, 2, 15 and 16. The "Load Shed Value" property sets a load shed value. The user can assign a value of 1-8 on any output. The "Sequencing Property" may be used to assign a sequencing level to an output that draws several amps. Values of 1-4 may be assigned to any output. A 1-4 second delay is implemented for turning on or off an output. The "Diagnostics" property sets may be set to true or false. It should be set to false if the output is driving a relay or device that is drawing less then .75 amp. The "Capacity" property defines the capability of this channel in amps and should not be exceeded. The "Bound To" property reflects the user's design decision to enslave two or more outputs together. Finally, the "Pin" property identifies the connector pin for this output.

Outputs and Relationships

The V-MUX System Designer enables the user to define the operation of a vehicle network by assigning commands in the Relationship window for output - unless, of course, an output has been defined as True or Permanently on.

In order to properly define the operation of a vehicle network, the user must know the desired effect of an input or set of inputs. The user then selects the appropriate command(s) to accomplish the desired effect from the command database.

Referring now to Figure 13, there is illustrated an example Relationships window. Notice that Water Temp High is in the lower portion of the window 1312 by itself. The user can move a command from the upper portion of the window 1310 to the assignment lower (assignment) portion of the window 132 in one of two ways. The user may click once on the command then once on the appropriate operand, or click and drag the command into the lower window the operand box will prompt the user for a choice. The operands are very easy to use. The "ON" operand is assigned by default for the first command selected.

The "AND" operand is used when more then one condition must be met to turn on an output. For example, Water Temp High AND Park/Neutral requires that the water temperature to be above the high threshold and the vehicle must be in Park/Neutral for the command to be sent. The "OR" operand is used if an output is to come on with either one of the two commands selected.

The "XOR" stands for exclusive OR and is used when one or the other command may be use to change the status of the output but not both. The XOR may conveniently be used for three way switching when two different commands are used. For example, if a vehicle design required a light to turn on or off from two locations, but the switches also served two other functions, then it would be appropriate to use the XOR operand. To control a device from two locations with the same command, the user must set the three way property to "true" in the Input properties for the inputs assigned to the command. The NOT operand is use to set a condition based on an input being off instead of on. For example, NOT Park/Neutral would turn on an output when the transmission is not in park or neutral. An output can control virtually any attached device as long as the device is not drawing too much current.

Adding a Vista Display Node

Referring now to Figure 14, there is illustrated a sample Design Tree having a Vista display node entry defined as node 6 1410. A Vista display node may be added to a Design by selecting the Vista icon from the tool bar. By default, Command Sets and Screen 1 are added. As shown in Figure 14, Node 5 has three screens and Node 6 has the default settings. Screens can be added to a Vista display node by clicking on the screen icon on the tool bar. The icon is ghosted out, if the Vista has not been selected. Several of the icons on the tool bar are only selectable if the proper item is selected in the design tree that correlates with the icons on the tool bar.

Screen Properties

Referring now to Figure 15, there is illustrated the Properties window for screen 1 of Vista display Node 6. Naming screens before setting the Vista properties will be more intuitive for most users. As shown, the Name property can be changed to, for example, "Main Menu". The next property is the Screen Type. There are three types of screens: Menu, Information and Diagnostics. The Menu is the most common screen. It allows for commands and tittles to be assigned to each button, just like inputs. The Menu screen also allows for three analog data values to be shown with their names. In the preferred embodiment, the Information screen is a pure text screen used for displaying vehicle information, creating "Splash" screens, or providing other useful text-based information. The Diagnostics screen is helpful in trouble shooting the V-MUX system and provides useful diagnostic information. Title Line 1 will use the default title assigned in the main Vista properties if it is left blank. Title Line 2 is displayed directly below

the main title, and it may be used to name screens so that when the end users are traveling through menus they don't get lost.

Properties of the Vista Display Node

Referring now to Figure 16, there is illustrated the properties window of Vista display node 5. The "Name" property of a Vista display node is automatically assigned by the system and cannot be changed by the user. The "Node Type" property is assigned when the user selects the Vista icon from the tool bar. This also cannot be changed by the user. The "Default Title" can be changed by the user, and may be up to 40 characters in length including spaces. The default title will show up in every screen that the user has not defined the Title Line 1 property. The "Home Screen" property identifies the screen to display at startup and when selecting the home button. The "Splash Screen" property enables the user to create a splash screen by adding a screen to the Vista and setting the screen property for that screen to "information".

The "Alert Beep" property enables the user to select from several beep styles to alert the end user of a System Error Message. The "Button Beep" property enables the user to select from several styles of button beeps every time the end user presses a button, the Vista node will beep according to the selected style. The "Multi-Level Delay" property enables the user to select from ten levels of delays. This allows the user to delay sending a command when using PWM control. "Show Analog Warnings", "Show Diagnostics" and "Show Loadshed" properties are all True or False options. Set to true to display messages on the 14th line of the Vista. "X location" and "Y location" are properties that reflect the location of the node for the Layout report. "Library Space" indicates how much space is used and how much is free. The example indicates that 0 bytes are used and 10240 bytes are free.

Command Sets

Referring now to Figure 17, there is illustrated an exploded Design Tree including the command sets of nodes. The V-MUX System Designer enables a user to build sets of commands in each display and I/O node. The user may assign one or more command sets to the inputs or buttons of the associated display and I/O nodes. A command set may be used when it is necessary to send more then one command from an input or button. By default two command sets are included with a Vista display node, but this can be increased up to 16. There may be up to 20 commands to each command set.

To preview or define the properties of a command in a command set, the user may select the command and right-click. This will cause a list of command categories to be displayed, similar to the categories displayed during the assignment of inputs. Selecting the appropriate category and the command to be assigned causes the Properties window to be displayed. Figure 18 illustrates the properties window for command 1710 shown in Figure 17.

In most cases, only a few commands will be used in each command set. The present example illustrates the implementation of an emergency master switch which allows the end user to press one button to cause the node to send out all the necessary commands to turn on all the emergency devices. In the past, end users may have used latch switches and left them in the latched position. Then wired the emergency master switch to power all these other switches. This is the same idea, except there are no switches to wire. One command set is defined for ON and a second command set is defined for OFF.

As shown in Figure 18, the "State" property for the Strobe Front, Command 1, is set to On. This means that when this command set is executed the command Strobe Front On will go

out to the V-MUX system. The "Extended" property is set to False. This three-way property and would be set to True if the design required the status to be the reverse of the last time this command was sent. Commands sets can also be assigned to a Vista Screen under the On Entry Execute property. One useful feature is the ability to assign a relationship to each command set. For example, by clicking on Command Set 1, the user is able to drag a command from the top half of relationships window to the lower assignment half, just like an output of a node. The user should use this property to prevent commands from being re-executed for some reason.

Menu Screens and Button Properties

Referring now to Figures 19 and 20, there are illustrated an exploded portion of the Design Tree showing entries for Emergency Master button 1910 of the EMM Screen 1912 selected, and the associated Properties window for the emergency master button.

As shown in Figure 4, there are 12 buttons on a Menu Screen. Buttons 1-4 (412-418) are at the left side of the screen, buttons 5-8 (420-426) are on the right and buttons 9-12 (428-434) are along the bottom. As shown in Figure 19, all of the buttons for the EMM Screen 1912 have been assigned descriptive names. Figure 20 illustrates the properties of button 11, the emergency master button 1910.

The "Name" property is derived from the Command Property. There are three "Button Types": Switch, Screen Link and Multi-Level PWM. The "Behavior" property may be one of a set of values. The default value of "Virtual" will be appropriate in most cases. Using the Virtual behavior property will reflect when the command assigned to this button is used from other locations.

The other behavior property is Latched which means the legend assigned to this button (ON/OFF), can only be changed by this button. The "Command" property is used to assign a command to a button just like the inputs on an I/O node. The "Displays" property may be one of several variations of the ON/OFF. "Off Execute" and "On Execute" are two very powerful properties of a button. The user can assign Command Sets to these two properties to send multiple commands across the system. In present example. Command Set 1 is assigned to "On Execute" and Command Set 2 is assigned to "Off Execute". The V-MUX System Designer allows the user to define multiple commands to be processed, for example, all the commands which may be turned on and off by just pressing the Emergency Master button. Accordingly, a single button can result in up to 20 ON commands, to be sent over the V-MUX bus. Likewise, up to 20 OFF commands could be sent.

Referring now to Figure 21, there is illustrated an alternate Properties window for Emergency Master button 1910. The "Button Type" property is "Screen Link". This directs the display node to display a linked screen when the Emergency Master button is pressed. The "Link" property identifies the screen to which the button is linked. As shown, the user may select the screen from a drop-down list of defined screens 2110.

Multi-Level PWM Button Type

The Multi-Level PWM button is a useful button type, enabling a user to define a set function for a button feature. For example, a button can be programmed to act as an incremental dimmer. Referring now to Figure 22, there is illustrated a design tree including a work menu 2210 and a dome lights button 2212. As shown, the "Button Type" property is assigned "Multi-

Level PWM," causing many of the properties to be assigned "N/A." The four lower properties apply to multi-level PWM buttons, and can now be used.

The "Levels" property allows the user to set the number of PWM levels assigned to the button. In this case, the user chose 10. The Design Tree of Figure 22 reflects the number of levels selected. The "Node" property identifies the node to use the channels on. The last two properties "Channel A" and "Channel B" allow the user to select from 1-4 the PWM channels controlled on the selected node. Logical Channels 1 and 2 are mapped to physical channels 1 and 2 and logical. Channels 3 and 4 are mapped physical channels 15 and 16 in an I/O node.

Once all the properties for the button have been set, the user may set any interlock for that button by dragging it down from the relationship "Commands Available" window to the relationship "Assignment" window.

The properties for each level may be set by the level as shown in Figure 24, and filling in the properties as shown in Figure 25. The "Display" property may be assigned any text that makes sense. In most cases, the user should think in terms of levels, and assign, for example, "Level 1" or a word like "low." The "PWM Percent" property indicates the percentage of power supplied to the output for the button step. It is important to define at least one level that is "off", where desired. The PWM controls will control an output on any I/O node and will override any relationships other than the one assigned to the button. For example, if the Side Door has been assigned to turn on the dome lights at 50%, the Multi-Level PWM control on a display node will take over until the control has been turned to Off.

<u>V-MUX Downloader</u>

In the preferred embodiment, the V-MUX downloader requires the user to Plug a Duetsch four-pin communications connector from the V-MUX Transceiver into a four-pin mate on the vehicle. The nine pin serial connector must be connected to the pc that has the V-MUX Downloader software on it. Once all connectors are complete, a green led light toward the center of the transceiver will be illuminated when the power is on.

To download a file, the user simply clicks on button 2610 on the top right, this allows the user to browse the computer to retrieve a stored binary file. Next, the user selects the communications port to which the nine-pin serial connector is connected to. The user may simply slide bar 2612 from left to right. Typically com1 will be used. Next the user must choose whether the node is blank or active. This can be determined by examining the end of the node when it is powered with 12v dc. If the green light is flashing, then it is an active node. If the green led is solid then it is a blank node. Next the user may click the download button 2614 to begin downloading the data to the node. During the download, the green led should be on solid, even if it was flashing to begin with. Also, a red led next to the green one should be solid indicating the node is receiving data over its comm port. When the status bar 2616 is near the end, observe the green led, it should pulse a few times to indicate the end of the download procedure. If the status light does not pulse, the node may not have properly received all the data.

From the above description of the invention, those skilled in the art will perceive improvements, changes and modifications in the invention. Such improvements, changes and modifications within the skill of the art are intended to be covered by the appended claims.

Accordingly, it is to be understood that the drawings and description in this disclosure are proffered to facilitate comprehension of the invention, and should not be construed to limit the

scope thereof. It should be understood that various changes, substitutions and alterations can be made without departing from the spirit and scope of the invention as defined solely by the appended claims.